Overview
- Overview of a lifecycle or development process
- Tools used in each phase
  - Design
  - Development
  - Test/Stabilization
  - Deployment

Why have a process?
- **PRO:** Methodology tells you how to do something
  - Seems like a good idea...
  - Carries lessons learned forward so we don’t repeat mistakes
  - Makes sure we’re thoroughly understanding the problem
- **CON:** Most people do not have the experience to know which parts of the method help get them to the goal
  - Methodology can lead to “analysis paralysis”
  - Methodology cannot solve fundamental people issues like communication
- What’s your methodology? How do you do team projects?

Lots of processes exist…
- Mil-Std 2167A, 10000 series…
- Capability Maturity Model (SEI)
- PMBOK (IEEE)
- Extreme programming (XP), agile methods
- “Corporate custom”
  – I’ll use Microsoft Solutions Framework terminology
  – Most methods have similar steps but often call them by different names

Process Model for Application Development

Milestone-Driven Process
- Milestones are review and synchronization points, not freeze points
- Milestones enable the team to assess progress and make mid-course corrections
- The process model uses two sorts of milestones
  - Major milestones
  - Interim milestones
- Achieving a major milestone represents team and customer agreement to proceed
- Deliverables are physical evidence that the team has reached a milestone
Milestones

- 1/23: Teams: Progress Reports
- 1/26: Teams: Technical Specifications
- 2/01: Teams: Technical Specifications due by 3 pm
- 2/07: Project—
  - A webpage that will control a QuickTime instance, play, stop, alert current position
  - Example PHP scripts that store and recall data from the MySQL database, movie title, clip URL, clip start time, clip stop time
  - Have a PHP script that generates an arbitrary SMIL file given dummy data
- 2/09: Have a plan for final UI
- 2/10: Project—
  - Webpage JavaScript interface for POSTing to PHP program
  - PHP script for generating SMIL file now reads from database
  - PHP script to swap positions of video clips
- 2/15: Teams: Progress Reports
- 2/20 - 3/01: Teams: Prototypes
  - Combine previous projects into a prototype

Drive the architecture

Scope Complete Milestone

Signals agreement on
- The planned feature set
- Whether the planned feature set has been developed
- Baseline materials to support user performance
- The stabilization process, including betas and testing

Team Focus During Developing

<table>
<thead>
<tr>
<th>Role</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product management</td>
<td>Customer expectations management; communication plan execution; beta planning</td>
</tr>
<tr>
<td>Program management</td>
<td>Project tracking; team communication and coordination; beta planning</td>
</tr>
<tr>
<td>Development</td>
<td>Feature development; testing</td>
</tr>
<tr>
<td>User education</td>
<td>User performance support development and testing, beta planning; product usability testing</td>
</tr>
<tr>
<td>Testing</td>
<td>Test specifications, cases, and scripts development; testing</td>
</tr>
<tr>
<td>Logistics management</td>
<td>Operational support documentation; beta planning; internal team support</td>
</tr>
</tbody>
</table>

Internal Releases

Getting the product to a known state and incrementally building upon it
Daily Build

Building the product in an executable form on a daily basis

A public daily build is
- A strong indicator that a team is functional
- A way to make the product and its progress visible
- The heartbeat of the development process

Video

Release Milestone

Signals agreement on
- Product stability and resolution of all known bugs
- Customer acceptance of the product
- Transfer of ownership for long-term management and support
- Change in team focus to the next release

Suggested Interim Milestones

Entering the stabilizing phase marks the transition from a schedule-driven focus to a ship-driven focus

Focus on Shipping

Beta Testing

Testing of a stabilized product by external end users
- Provides actual end-user usage testing in the expected environment
- Requires greater team effort than alpha testing
- Occurs with different frequency and size depending on a number of factors

Bug Triaging

Evaluating and prioritizing bugs to determine their appropriate resolution
- Uses a review committee to prioritize and assign bugs
- Determines what new bug fixing, if any, will be done
- Balances stability against customer needs
- Can result in loss of features for the sake of stability
Tools we use….

Design tools
- Word, Excel (noun analysis, contracts)
- Visio (UML and ORM), Erwin, Rational XDE
- MS Project (schedule)
- Team portal and bug tracker set up
  - SharePoint, raw HTML, FrontPage
  - TaskVision
- Whiteboard…

Development tools
- (your compiler 😊)
- NAnt, Ant, Kinook (builds)
- FxCop (secure code reviews)
- Visual SourceSafe, PVCS, Clearcase (code mgmt)
- NDoc, HTMLHelp, Robohelp (documentation)

Test and stabilization tools
- VSS, Clearcase, PVCS
- NUnit, NUnitASP, JUnit
- App Center Test
- WinRunner, XRunner
- NDoc, HTMLHelp, Robohelp
- FxCop
- MSI or InstallShield
- CLR Profiler

Show me the code!
- Testing a simple customer app

BACKUP SLIDES
Testing in development

- Coverage testing
  - Attempts to thoroughly test every feature of the product
  - Attempts to thoroughly test the code base of the product
  - Is used primarily during the developing phase
- Usage testing
  - Attempts to successfully complete usage scenarios
  - Attempts to test the product in its expected environment as users might stress it
  - Is used primarily during the stabilizing phase

Types of Testing

- Unit tests
- Functional tests
- Check-in tests
- Build verification tests
- Regression tests
- Configuration tests
- Compatibility tests
- Stress tests
- Performance tests
- Documentation and help file tests
- Alpha and beta tests

Zero-Defect Mindset

Committing to the highest possible level of quality within project constraints

- Team members must understand the required quality level for their work
  - Articulate the quality bar for all work performed – who does this?
- Work is not complete until it reaches that level of quality
- The zero-defect mindset is embodied in
  - Task deliverables
  - Milestones

Benefits of a Zero-Defect Mindset

- Increases accountability for the quality of the product
- Increases stability of the product
- Improves schedule predictability
- Decreases the cost of addressing issues
- Allows testing to shift focus to quality assurance
- Rewards quality developers

Techniques for Zero-Defect Development

- Write unit test cases before debugging
- Assume the code is broken, then prove that it isn’t
- Fix bugs before moving on
- Use competing designs and implementations
- Assign bugs to other developers
- Reassess the code in light of bugs
- Document code
- Conduct code reviews
- Perform daily builds

Code Reviews

Assessing code to improve its quality and to improve the capabilities of the development team

Some ways to conduct code reviews
- A comprehensive, formal review
- A more casual, peer-based review
- An independent, third-party review
- Tools plus people...
- Daily build status and source control are evidence of a working process
Guidelines for Internal Releases

- Treat internal releases within a single project like versioned releases of a product
- Address high-priority and high-risk features in the earliest possible release
- Define a quality bar to determine when the product has met the standards for internal release
- Make each release as cohesive and yet independent as possible
- Conduct postmortem reviews of each internal release